



## *Investigating Delay-Power Tradeoff in Kogge-Stone Adder in Standby Mode and Active Mode*

Design Review 2, VLSI Design – ECE6332  
 Elaheh Sadredini  
 Luonan wang

November 11, 2014

### 1. Research

In this design review, we investigate power and delay tradeoff in Kogge-Stone adder and basic carry look-ahead adder. We have read some papers on parallel prefix adders and also some leakage reduction techniques. Kogge-Stone adder has an interesting structure and gives us some symmetric pattern to investigate power reduction techniques.

### 2. Design and Simulation

We started with designing and simulating for 8bit and 16 bit Kogge-Stone adder (KSA) and carry look ahead (CLA).

Kogge-Stone Adder:

KSA has three component: 1. PG generator, 2. Dot block, and 3. Sum generator [Appendix B]. The number of dot blocks are more than PG generator and sum generator. So we call it dominant block when talking about leakage power consumption of the CSA. Table 1 present leakage power for all possible input vectors.

P1	P2	G1	G2	Leakage Power (nw)
0	0	0	0	41.51
0	0	0	1.1	44.02
0	0	1.1	0	55.53
0	0	1.1	1.1	42.77
0	0	0	0	60.04
0	0	0	1.1	51.78
0	1.1	1.1	0	50.66
0	1.1	1.1	1.1	39.77
1.1	0	0	0	51.02
1.1	0	0	1.1	48.53
1.1	0	1.1	0	60.04
1.1	0	1.1	1.1	47.28
1.1	1.1	0	0	70.72
1.1	1.1	0	1.1	62.46
1.1	1.1	1.1	0	61.34
1.1	1.1	1.1	1.1	50.45

Table 1. Dot block leakage power for all possible input vectors

P1 P2 G1 G2 = 0000 (I1) and P1 P2 G1 G2 = 0111 (I2) present the lowest leakage power for the dot block. We want dot blocks

have these inputs while the circuit is in standby mode. If we apply I2 to the last level of dot blocks and trace back towards the inputs, we see that it is not possible for all the dot blocks to follow I1. It is possible to come up with some block which their input pattern create a high leakage power. In contrast, if we use I2, each dot block can have I2 as its input and as a result, all the primary inputs of the adder are zero (symmetric input pattern). We concluded that having all inputs to zero have the lowest leakage power for the CSA and the simulation results (Table 2) approve it.

Input vectors	Leakage Power (uw)	
	8 bit KSA	16 bit KSA
All '0'	1.89	4.48
All '1'	1.96	4.61
101010...1010	1.96	5.97

Table 2. Leakage power for some input vector (8 bit and 16 bit KSA)

In the next step, we used three power gating techniques (A NMOS footer, a PMOS header, both header and footer) with setting all inputs to '0'. Then, we simulated all three techniques with different 'widths' for sleep transistor and calculated leakage power and propagation delay for 8 bit KSA.

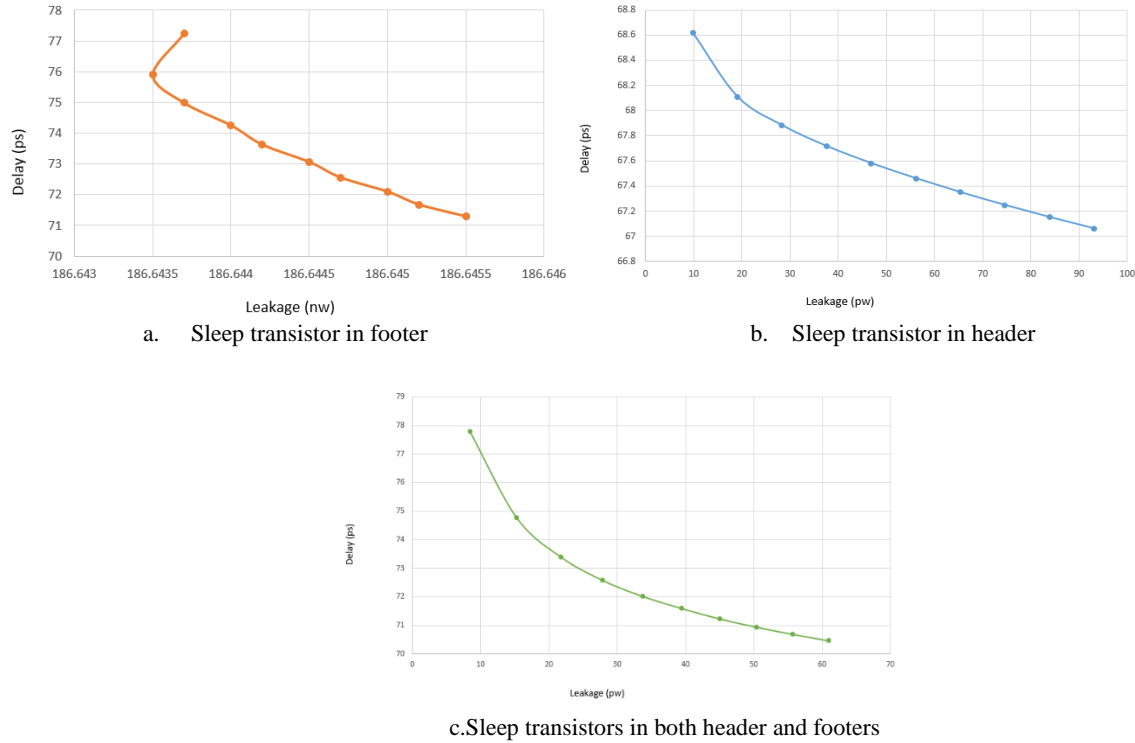


Figure 1. Leakage-power tradeoff for different power gating techniques with different sleep transistor width (w= 50n, 100n, 150n, ..., 500n)

Figure 2. CLA

We observed that when we use just footer as a sleep transistor, the leakage power reduction is about 10x, but when we use header or header-footer, the order of leakage reduction is about 1000x. It happened for both KSA and CSA. This result for footer is kind of unexpected. We still have not figured out why it happens.

### 3. Power reduction in active and standby mode for KSA

From Table 1, we figured out that for the input vector which is P1 P2 G1 G2 = 1100 (I3), leakage power is worst. We wrote a program with SystemC (why SystemC? It is illustrated in Appendix A) that calculates the possibilities of having each possible input vectors for each Dot block. Figure 3 shows that how often each dot block in 8 bit KSA can get I3 as its input vector. As Figure 3 represents, first level dot blocks have the most possibility to have I3 as its input vector.

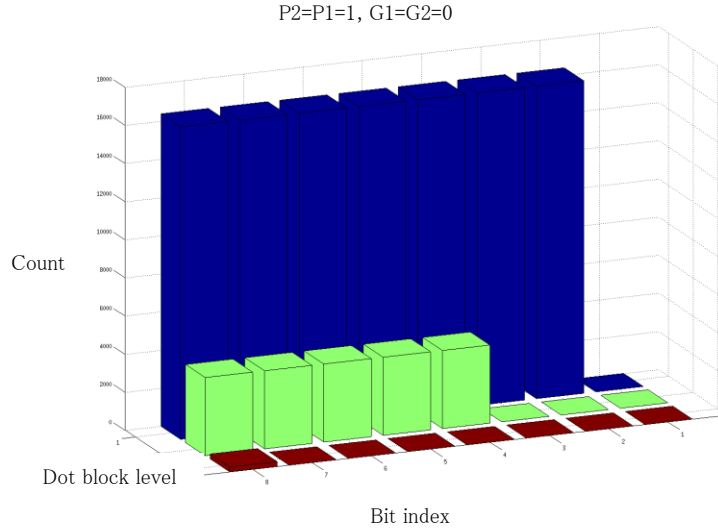


Figure 3. Number of having  $P1P2G1G2=1100$  for each dot block input in 8 bit KSA  
Generally, the basic idea here is that using a multi-mode power gating structure which is described in Figure 2 [2].

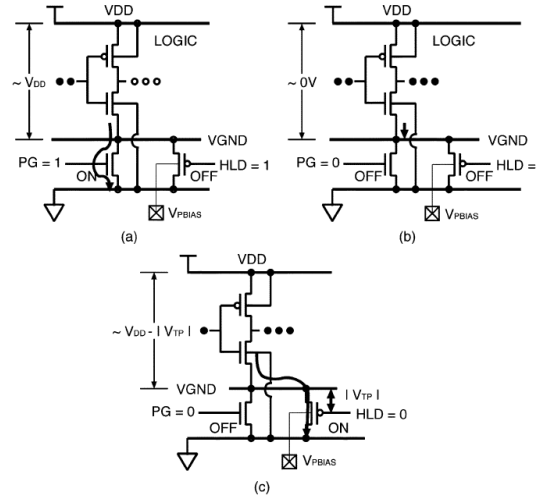
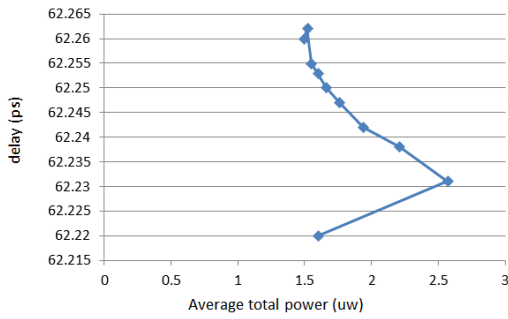


Figure 4. Multi-mode sleep transistors: a. Normal mode, b. Cold mode, c. Park mode (intermediate power saving mode) [2]  
When one level dot block is in active mode,  $PG=1$  and  $HLD=1$ . When one level dot block finished (circuit is still in the active mode), it can be put in the Park mode. It means that  $PG=0$  and  $HLD=0$  and in  $VGND$ , we have  $V_{thp}$ . The advantage of this mode is that it retains the output of the dot block (because next level dot blocks need its output) and because the biased voltage has decreased, the power consumption decreases. Finally, when the circuit is in standby mode,  $PG=0$  and  $HLD=1$  and leakage power will be reduced.

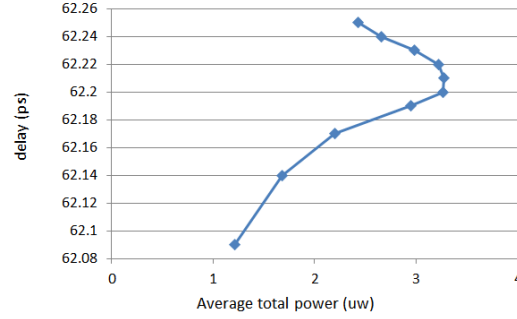
In 8 bit KSA, we want to use apply our low power technique to the first level dot blocks. There are three reasons for this. First, the simulation result shows that that first level KSA adder have more potential for consuming leakage power. Second, the number of dot blocks is larger in the first level and if we use our low power technique, it is more possible to save both dynamic and static power. Third, it is more feasible for the first level dot block to go to the Park mode, from the timing aspect and maintaining critical path. We examined three types of multi-mode sleep transistors in the first dot block 8 bit KSA (In PDN, PUN, and both) in PARK mode with different transistor widths. Table 3 show the delay and average power in the active mode for 8 bit KSA without adding the power gating technique.

Average power consumption in the active mode (uw)	28.72
Propagation delay (ps)	62.25
PG generator delay (ps)	22
Dot block delay (ps)	15
Sum generation delay (ps)	23

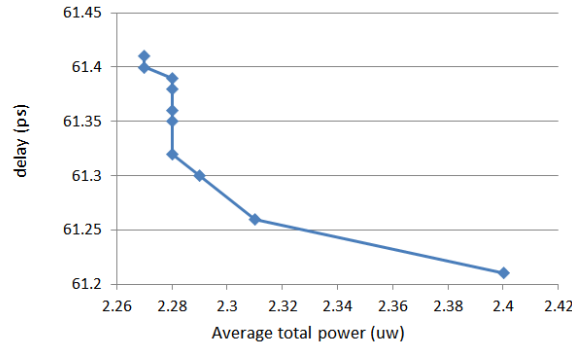
Table 3. Propagation delay and average power consumption in 8 bit KSA



a. Power gating in footer



b. Power gating in header



c. Power gating in header and footer

Figure 5. Power delay tradeoff in Park mode (first level dot block) for 8 bit KSA

As we, the total power consumption has decreased. We need to adjust this decrement to the portion of one level dot block delay to report the actual power reduction. In the future, want to simulate it in the standby mode to see how much leakage power reduction has gained and see how much our method can be effective.

There are two main concerns with this technique. First, we have to generate a clock which is faster than system clock and second, when switch from active mode to the Park mode. We want to investigate them, too.

#### 4. Progress, remained tasks

Progress table:

Task	who
Drawing the schematic of 4 bit, 8bit, and 16 bit Kogge-Stone adder	Elaheh
Writing SystemC code for calculating possibilities in dot blocks and ...	Elaheh
Simulating KSA with different techniques and different width using Ocean	Elaheh
Power reduction technique in active and standby mode for KSA	Elaheh
Drawing the schematic of 4, 8, and 16 bit carry look ahead adder	Luonan
Simulating CLA with different techniques and different width using Ocean	Luonan
Paper Summarize	Elaheh and Luanon
Writing Design Review 2	Elaheh
Creating wiki page	Luonan

Remained tasks:

Task	who
Drawing the schematic of 32 and 64 bit Kogge-Stone adder	Elaheh
Developing the idea of variation (if it was possible) with SystemC simulation	Elaheh
Simulating KSA with different techniques and width using Ocean for 32 bit and 64 bit	Elaheh
More investigation on multi-mode power gating technique technique for KSA	Elaheh
Drawing the schematic of 32 and 64 bit carry look ahead adder	Luonan
Simulating CLA with different techniques and width using Ocean for 32 bit and 64 bit	Luonan
Have some comparison with the previous works	Elaheh and Luonan
Solving the problem which we faced during design review 2	Elaheh and Luonan

## 5. Challenges and Question about the proceed

Challenges:

1. Working with ocean script and doing simulation got some time in the beginning. Facing with some wired errors. Finally, we made it work.
2. It is very time consuming using ocean to calculate every combination of input for more than 8 bit adder, so we wrote a SystemC based simulation to calculate possibilities of the inputs of the dot block and the estimate leakage power based on dot block leakage power for different inputs. For example for 8 bit adder, ocean takes around 8 days to be completed
3. For the kogge stone adder, it is very time consuming to create 32 or 64 bit adder. Because we cannot use for example two 16 bit KSA adder to create 32 bit KSA.

Questions about proceed:

1. If we will get glitch from the process variation in the Kogge Stone Adder? If it happens or not?
2. Why when we use just footer (in Figure 1), the leakage is bigger than having header or header and footer. It happened for both KSA and CLA.
3. Why there are some unexpected results in Figure 5-b.

## 6. Paper summary

### A New Optimized High-Speed Low-Power Data-Driven Dynamic (D3L) 32-Bit Kogge-Stone Adder

Using fast design logic style, such as Domino Logic, can improve the adder speed. However, despite the high speed reached by a Domino Logic parallel prefix adder, such a circuit dissipates a large amount of energy due to the presence of the clock distribution tree which inputs the clock signal to all the logic gates.

Data Driven Dynamic Logic (D3L) achieves a considerably energy saving, over conventional Domino Logic, by removing the clock signal: the control of the precharge and evaluation phases is managed only by input data. As a consequence, the power consumption is significantly reduced at the expense of a non-negligible penalty in terms of speed performances.

In an n-type (p-type) D3L gate, the clocked precharging PMOS (NMOS) transistor employed in Domino Logic is replaced by a Pull-up PMOS (Pull-Down NMOS) network, which receives a subset of the input data signals (the so-called pre-charge inputs) instead of the clock signal.

The evaluation network of the gate remains unchanged, with respect to the equivalent Domino gate, and the clocked NMOS (PMOS) foot transistor is avoided. The precharge inputs need to satisfy the following conditions: 1) during the precharge phase, the Pull-Down network (PDN) is OFF, the Pull-Up network (PUN) is certainly turned ON and the output node is charged to V<sub>dd</sub>; 2) during the evaluation phase, the output node is eventually discharged to 0 by the PDN without any contention with the PUN.

In this paper, a new parallel-prefix structure is presented to efficiently exploit D3L in the design of low-power high-performance adders. Moreover, a new dynamic design style, named Split-Path D3L, is proposed to overcome the speed limitations of traditional D3L. When applied to the design of a 32-bit Kogge-Stone adder, the proposed approach halves the precharge propagation path with respect to the traditional D3L design style, and allows a smaller sizing of the precharging PMOS transistors. As a consequence, the new technique leads to an Energy-Delay Product 25% and 20% lower than traditional domino and D3L logic styles.

### Resource Allocation and Binding Approach for Low Leakage Power

Static power dissipation due to sub-threshold leakage current in CMOS VLSI circuits becomes significant as current technology descend into deep sub-micron regime. Sub-threshold leakage current increases due to reducing threshold voltage to compensate for performance loss. To solve this problem, the author proposed a resource allocation and binding approach for low leakage power. A MTCMOS design style is introduced briefly which use a sleep transistor to isolate the circuit from the supply voltage and the ground rails during idle periods. Then an allocation and binding algorithm which attempts to maximize contiguous idle times of resources was proposed.

Because large modules contribute mainly to the performance loss, for example the multiplier, performance recovery based on multicycling and slack was proposed. They use total execution time as a metric, taking 6 control steps for each input vector, they were able to reduce the performance penalty from 40% to 14.28%. Then IIR filter was presented and leakage power savings in IIR with regular and multicycling were measured.

## 7. References

- [1] Rajani H.P., Srimannarayan Kulkarni, "NOVEL SLEEP TRANSISTOR TECHNIQUES FOR LOW LEAKAGE POWER PERIPHERAL CIRCUITS", International Journal of VLSI design & Communication Systems (VLSICS) Vol.3,

No.4, August 2012.

- [2] Suhwan Kim ; Seoul Nat. Univ., Seoul ; Kosonocky, S.V. ; Knebel, D.R. ; Stawiasz, K., “A Multi-Mode Power Gating Structure for Low-Voltage Deep-Submicron CMOS ICs”, IEEE Transactions, IEEE Circuits and Systems Society, Volume:54 , Issue: 7, July 2007.
- [3] Kumar, Y., Paliwal, S. ; Rai, C.K. ; Balasubramanian, S.K., “A novel ground bounce reduction technique using four step power gating”, Engineering and Systems (SCES), IEEE, 2013.
- [4] Benton H. Calhoun, Frank A. Honoré, and Anantha P. Chandrakasan, “A Leakage Reduction Methodology for Distributed MTCMOS”, IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 39, NO. 5, MAY 2004.
- [5] Anup Jalan and Mamta Khosla, “Analysis of Leakage Power Reduction Techniques in Digital Circuits”, India Conference (INDICON), IEEE, 2011.
- [6] Sathyabama University, Chennai, “Leakage Power Reduction in CMOS Modulo4 adder and Modulo4 Multiplier in Sub-micron Technology”, International Conference on Sustainable Energy and Intelligent System (SEISCON), 2011.
- [7] <http://venividiwiki.ee.virginia.edu/mediawiki/index.php/ToolsCadenceTutorialsBasic>
- [8] Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolic, “Digital Integrated Circuits: A Design Perspective”, Second Edition, 2003.
- [9] Chandramouli Gopalakrishnan and Srinivas Katkoori. “Resource Allocation and Binding Approach for Low Leakage Power”. Proceedings of the 16th International Conference on VLSI Design (VLSI’03).

## Appendix A

At first, we wanted to use our SystemC simulation to calculate best input vector for the circuit in the standby mode. Because it is much faster than Ocean. Besides that, we wanted to have some timing information (for the later usage maybe). To investigate 8 bit adder input vectors, ocean takes about 8 days and SystemC simulation about a couple of hours. Then, we used it to get information which is represented in Figure 3.

We also can use this simulation for variation purpose. Because of variation, it is possible to have some glitch in the circuit. By increasing transistor widths, the glitch effect can be alleviated. We want to use this simulation to see increasing transistor width in which part of the circuit can be more effective. We think that the parallel prefix adders which have more fan-out can be good case study.

## Appendix B

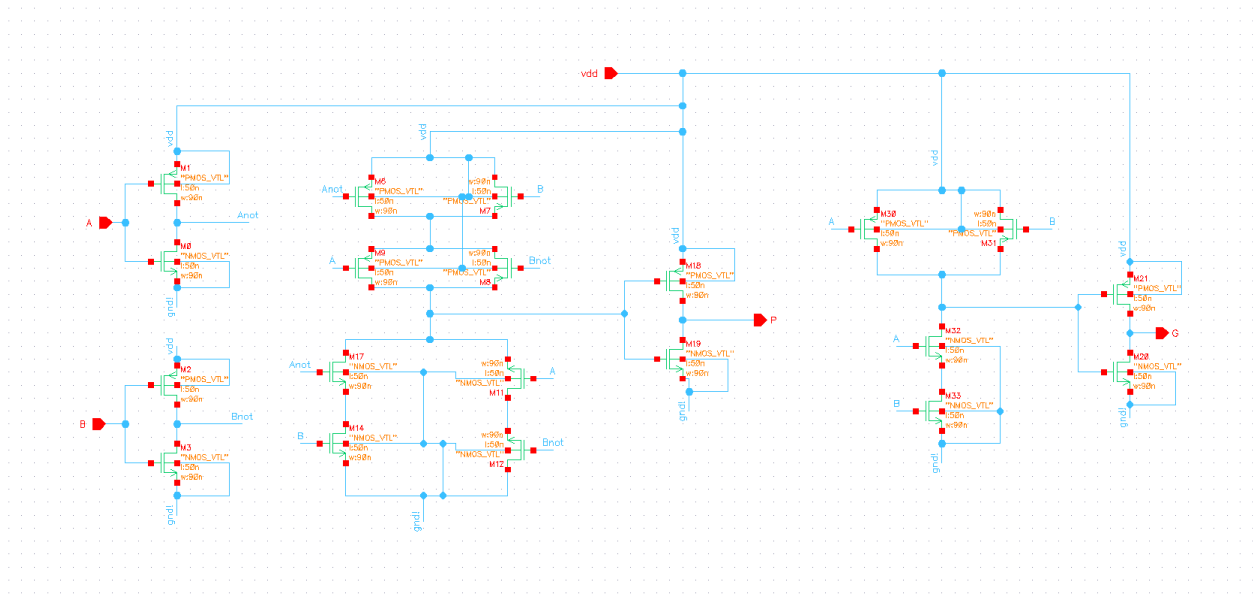


Figure 6. PG generator in KSA

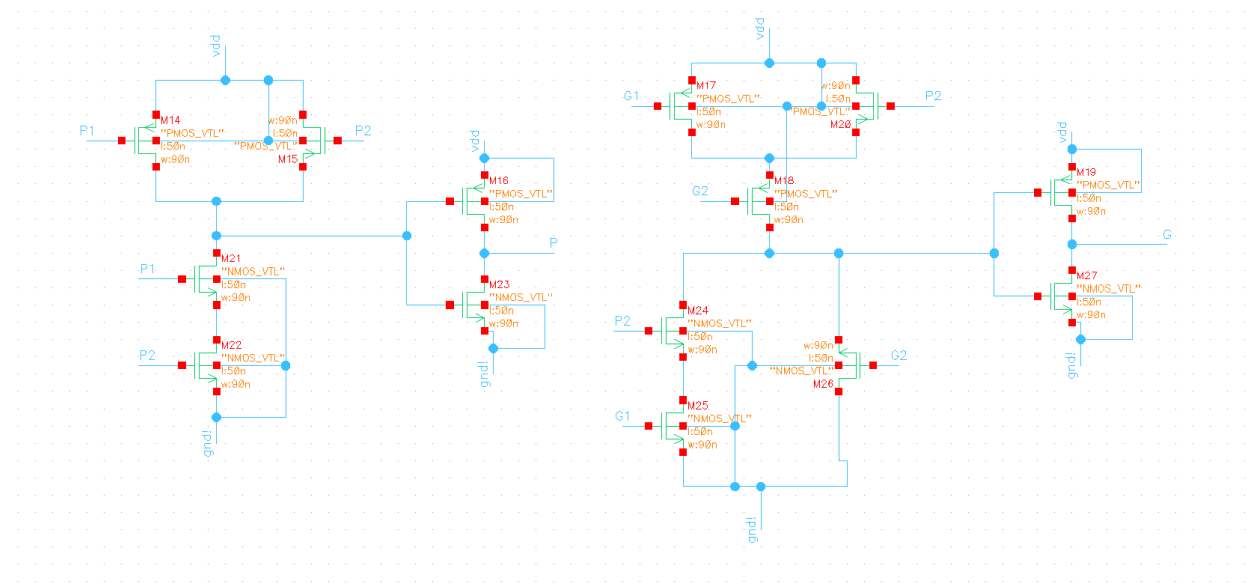


Figure 7. Dot block in KSA

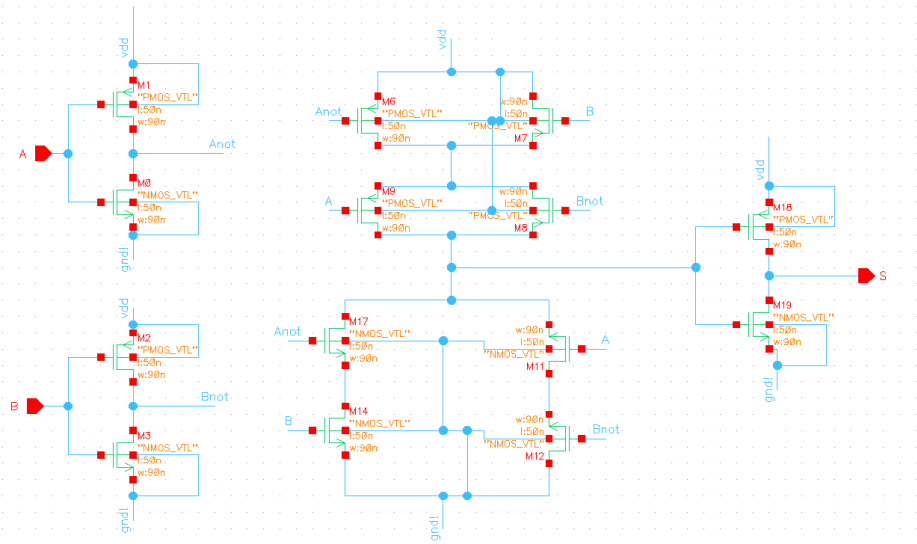


Figure 8. Sum generator in KSA

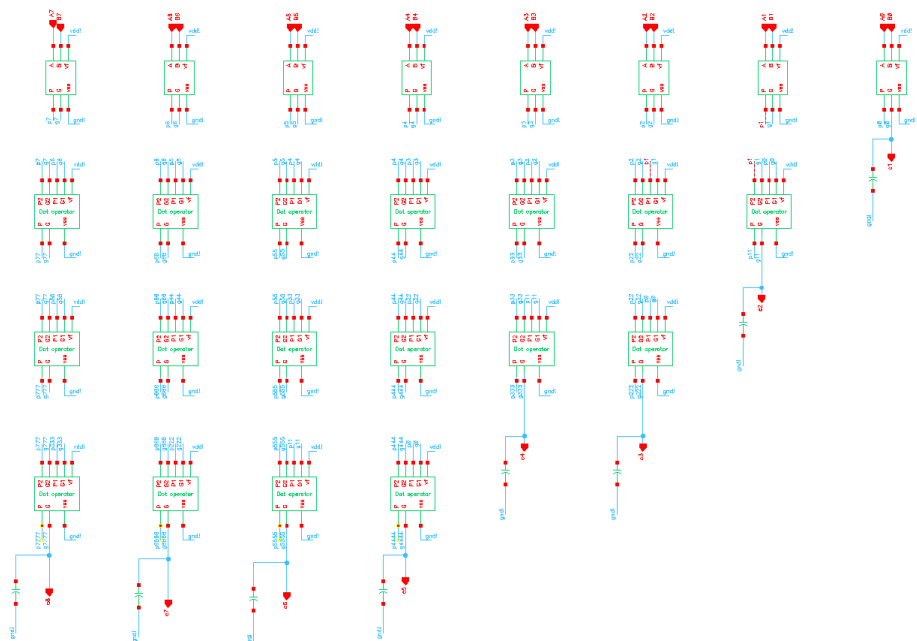


Figure 9. 8 bit KSA (Carry)



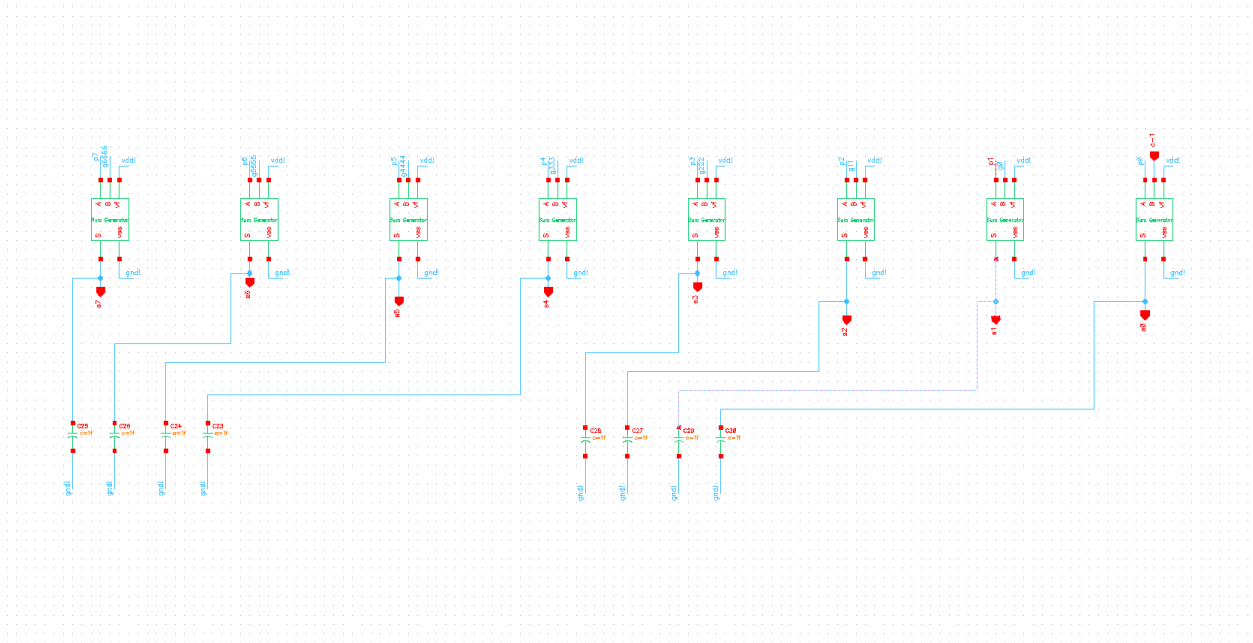


Figure 10. 8 bit KSA (sum)

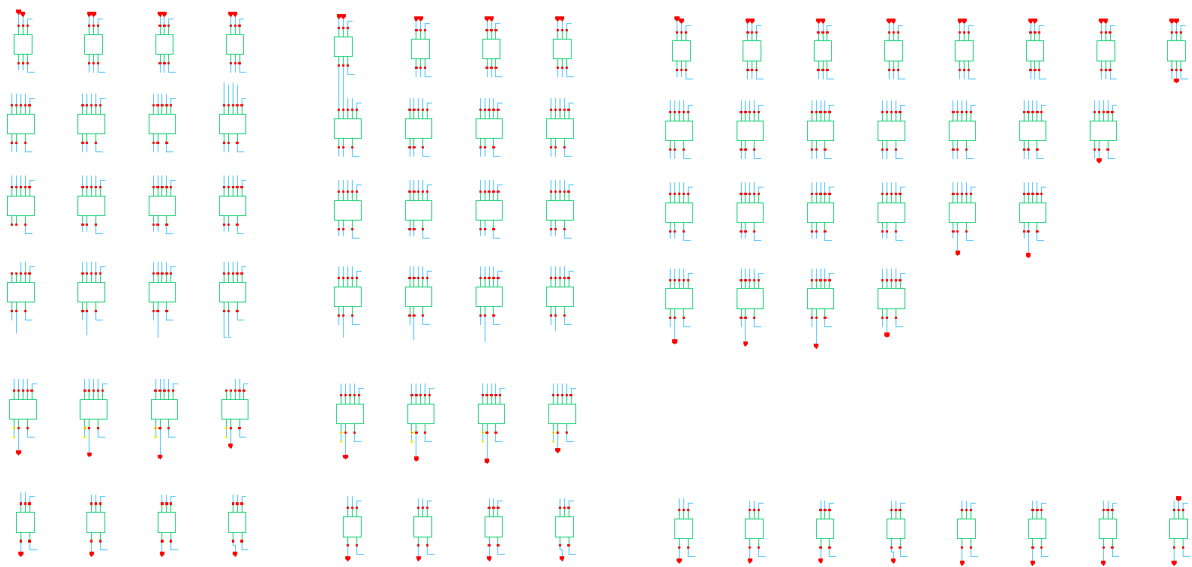


Figure 11. 16 bit KSA

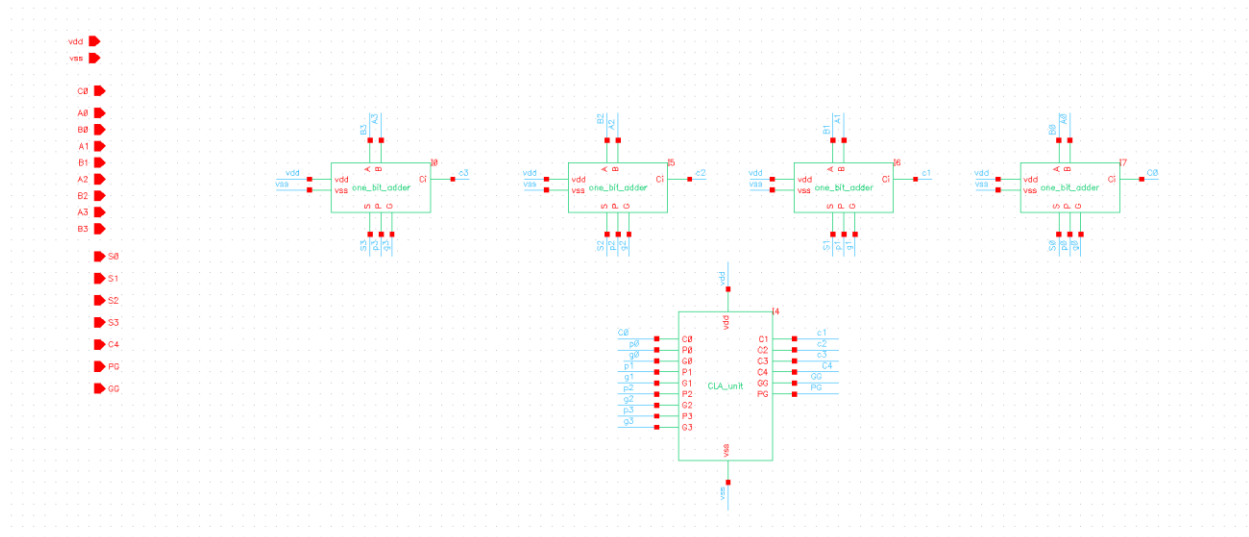


Figure 12. 4 bit CLA

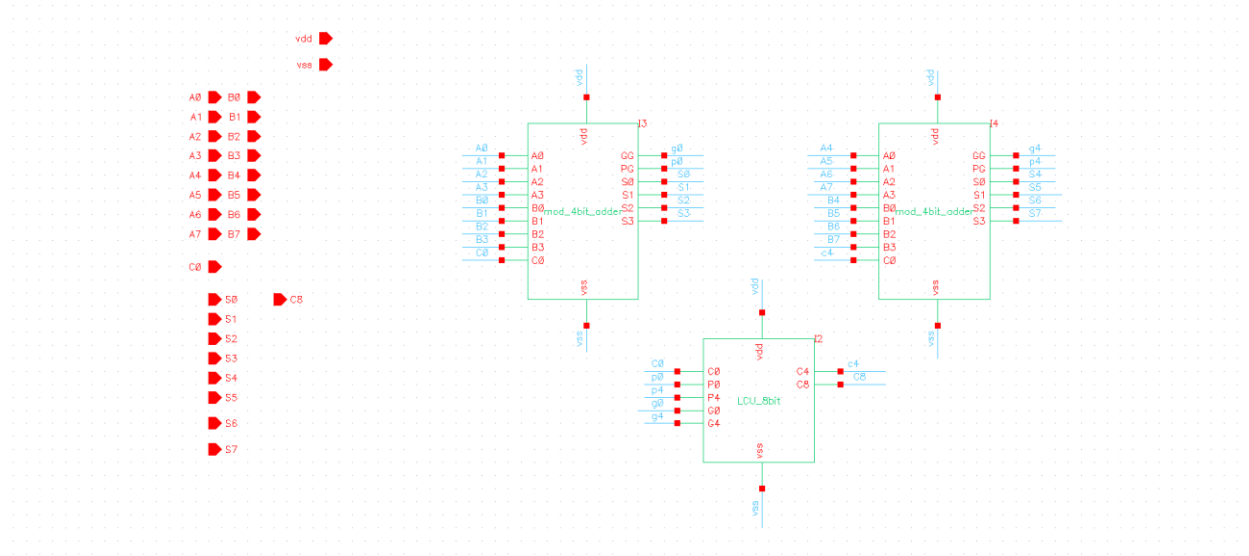


Figure 13. 8 bit CLA

